IXIR: A Statistical Information Distillation System

Michael Levit [a,*] Dilek Hakkani-Tür [a] Gokhan Tur [b]
Daniel Gillick [a]

[a] *International Computer Science Institute, Berkeley, CA*

[b] *SRI International, Menlo Park, CA*

Abstract

The task of information distillation is to extract snippets from massive multilingual audio and textual document sources that are relevant for a given templated query. We present an approach that focuses on the sentence extraction phase of the distillation process. It selects document sentences with respect to their relevance to a query via statistical classification with support vector machines. The distinguishing contribution of the approach is a novel method to generate classification features. The features are extracted from charts, compilations of elements from various annotation layers, such as word transcriptions, syntactic and semantic parses, and information extraction annotations. We describe a procedure for creating charts from documents and queries, while paying special attention to query slots (free-text descriptions of names, organizations, topic, events and so on, around which templates are centered), and suggest various types of classification features that can be extracted from these charts. While observing a 30% relative improvement due to non-lexical annotation layers, we perform a detailed analysis of the contributions of each of these layers to classification performance.

Key words: Question Answering, Information Distillation, Machine Learning, Information Extraction, Natural Language Processing

1 Introduction

We present an approach that, given a templated query and a collection of documents, selects those document sentences that are relevant to the query. The task is essen-

* corresponding author
Email address: `levit@icsi.berkeley.edu` (Michael Levit).

tial for template-based question answering (termed "Information Distillation" in the DARPA-funded GALE program (BAE, 2006)) where one is asked to find answers to queries from a possibly very large collection of documents. A finite number of query templates are agreed upon in advance and have variable slots that are filled with free-text descriptions at runtime. For instance, the template *"Describe the prosecution of* [PERSON] *for* [CRIME]*"* has two slots that could be filled with PERSON= *"Saddam Hussein"*, CRIME= *"crimes against humanity"*. One other example is the open-domain template *"Describe the facts about* [EVENT]*"* where slot EVENT might contain *"civil unrest in France"* or *"bird flu outbreak in China"*. The output of the information distillation system is a list of snippets (sentences or phrases) relevant to the query.

The online part of a typical question answering (distillation) system consists of the following stages:

1) *Query Processing* decides what kinds of information to look for in the documents
2) *Information Retrieval (IR)* retrieves documents that are likely to contain answers
3) *Snippet Extraction* uses the retrieved documents to select sentences (or parts thereof) that contain answers
4) *Answer Formulation* combines (ranks, removes redundancy and possibly modifies) the extracted snippets to form an output

In addition, offline document preprocessing can be used to simplify and speed up online processing. In this paper, we are mostly concerned with the snippet extraction (stage 3) and, in particular, with selecting relevant sentences from documents.

Several factors have influenced question answering research in recent years. The size of data sets has increased to reach about 500K documents in GALE (BAE, 2006). Starting with the European CLEF project (Magnini et al., 2003), multilingual and cross-language question answering began gaining importance. While finding answers in a source language remains an interesting option for future research, most current systems translate documents automatically and do question answering on the translated output. However, machine translation adds noise to the texts and makes the task more difficult. Moreover, when dealing with speech, automatic speech recognition results in additional noise.

While manually designed lexical patterns were shown to be very useful in extracting relevant sentences (e.g. Soubbotin and Soubbotin, 2001), the recent trends listed above suggest that robust, trainable statistical mechanisms should augment (if not entirely replace) pattern-based models. These methods are cheaper as they only require a minimum amount of human expertise (mostly yes/no relevance transcriptions) and better suited to deal with noise by learning from the actual data. For instance, Ravichandran and Hovy (2002) learn lexical patterns from the Internet, while focusing on precision, and our group reported earlier on a system that relied on a discriminative model to select relevant sentences for queries using word (and named entity) n-grams as classification features (Hakkani-Tür and Tur, 2007). The system was applied downstream of the University of Massachusetts INDRI search engine (Strohman et al., 2005) that retrieved relevant documents for a query. It showed a clear improvement in distillation F-measure compared to a baseline keyword spotting strategy.

The present paper extends this approach in several significant ways. First, our classification features are composed of not only word transcriptions and names, but also of other representations associated with the sentences. Among those we count syntactic parses, semantic predicate-argument structures, and various elements of Information Extraction (IE) annotations. By incorporating all these representations into one coherent structure, a *chart*, we are able to evaluate their contributions and combine them to optimize classification performance. Second, we also extract classification features from query slots. Eight query slot types were suggested for GALE Year-II evaluations: PERSON, ORGANIZATION, COUNTRY, LOCATION, CRIME, EVENT, TOPIC, DATE[1].

Our current approach treats slots that are named entities, as defined by Message Understanding Conference and (MUC) and Automatic Content Extraction (ACE) evaluation tasks (LDC, 2005), (such as PERSON, ORGANIZATION, COUNTRY) differently from the slots that do not (e.g., TOPIC and EVENT[2]), which can be *any* noun phrase, hence can be expressed with a much greater lexical variability than named entities (see the examples above).

We first reported on the classification advantages achieved through these enhancements in (Levit et al., 2007a). The present paper, being a full version of the previous publication, is intended to offer a formal justification for the statistical sentence extraction, explain details of the adopted classification approach such as feature extraction and determination of alternative names. We also explore individual contributions of various types of classification features to the overall system's performance and make an effort to place our approach in the context of other state of the art question answering systems.

The remainder of the paper is organized as follows: Section 2 formalizes a statistical model of sentence extraction in the question answering domain. In Section 3 we introduce the central notion of a chart, itemize annotation layers used for chart creation, and describe features that are extracted from charts including the *topicality features*, a special kind of slot-related features used to evaluate a sentence's relationship to query slots. Next, we present a series of distillation experiments designed to evaluate the contribution to classification of various features. We review related work in Section 5, and conclude the paper suggesting future directions in Section 6.

[1] Table 1 shows which of these types were relevant to the experiments described in this paper.
[2] Note that an EVENT in GALE has a different, much broader meaning than in ACE, and the two should not be confused.

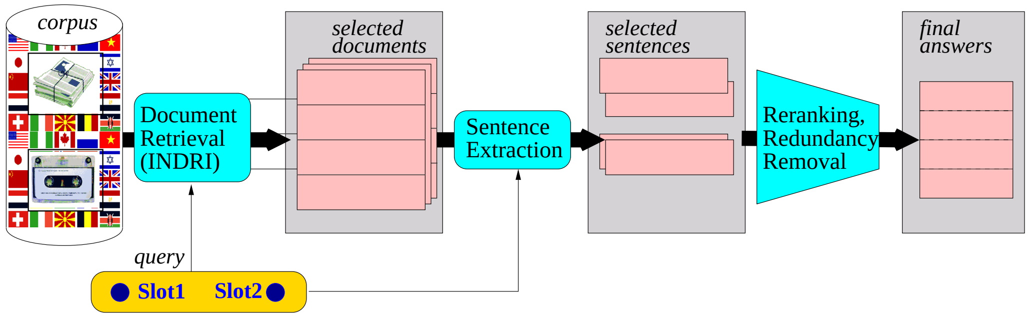

Fig. 1. IXIR distillation system

2 Statistical Sentence Extraction in IXIR

In accordance with the structural description of a typical distillation system presented in the previous section, IXIR's[3] general architecture, as shown in Figure 1, comprises an IR stage for document retrieval (INDRI (Strohman et al., 2005)), which is followed by a statistical sentence extraction mechanism and concluded by redundancy removal and snippet reranking performed at Columbia University.

In this paper, we focus on the sentence extraction module. To fully understand the extent of the problem, suppose that we need to find answers to a query about *prosecution of [al-Megrahi] for [the Lockerbie bombing]*, and one of the relevant documents from the IR stage contains the following paragraph (with sentence numbers shown for illustrative purposes only):

> (1) After months of testimony, judges on Thursday began deliberations on the fate of two Libyans accused of blowing up Pan Am Flight 103 in 1988 over Lockerbie, Scotland. (2) Lord Ranald Sutherland, the president of the court, said the judges would reconvene on Jan. 30, but would not deliver a ruling on that date. (3) The Libyan defendants, Abdel Basset Ali al-Megrahi and Lamen Khalifa Fhimah are accused of planting the plastic explosive in a radio cassette player on board the doomed aircraft. (4) Eleven people died on the ground and 259 in the air in the Dec. 21, 1988 blast. (5) If convicted of murder, Al-Megrahi, 48, and Fhimah, 44, would face life imprisonment in Scotland.

In this passage, a sentence extraction system must mark sentences (1), (3), (5) as relevant for the query, and the rest as irrelevant (references provided by human labelers).

Statistical methods of question answering and information distillation look at individual sentences s_i and reduce them to feature vectors $\mathbf{x}_i$ that are then used to train a binary decision function of relevance with respect to query q:

$$M: \ (\mathbf{x}, q) \rightarrow \{0, 1\}, \tag{1}$$

[3] IXIR stands for a free-style abbreviation of ICSI, SRI and Information Extraction.

where 1 means that the sentence is relevant for the query, and 0 means that it is irrelevant. In the definition of this function, the query q needs to be formalized as well. Where templates are involved, this formalization reduces the query to its template index t^q and one feature vector $\mathbf{y}_k^q$ per query slot:

$$M: \ (\mathbf{x}, t^q, \mathbf{y}_1^q \dots \mathbf{y}_K^q) \rightarrow \{0, 1\}, \tag{2}$$

where the number of slots K is usually between 1 and 3. This can be rewritten as a number of separate formulations for each template t^q:

$$M^q: \ (\mathbf{x}, \mathbf{y}_1^q \dots \mathbf{y}_K^q) \rightarrow \{0, 1\}, \ q = 1...Q \tag{3}$$

Now we can train Q statistical (discriminative) classifiers, one for each query template, that approximate decision functions M^q. Given the input feature vector for a sentence, $(\mathbf{x}, \mathbf{y}_1^q, \dots, \mathbf{y}_K^q)$, a classifier will output a score that, when above a certain threshold, will suggest that the sentence is relevant (class "1"). This threshold can be determined using leave-one-out optimization. In the approach we described in (Hakkani-Tür and Tur, 2007) a first step in this direction has been taken. Classification features comprised word n-grams and named entities from the sentences, while slots were given only a rudimentary treatment. The next sections focus on the algorithmic base of our latest, better-informed system (Levit et al., 2007a) that not only introduces slot features $\mathbf{y}_k^q$, but also greatly extends the dimension of the sentence-internal feature vectors $\mathbf{x}_i$. It should be noted that a fast sentence extraction mechanism would not require an upstream document retrieval stage. The main role of the typically computationally inexpensive document retrieval is to speed up distillation while improving precision owing to a more holistic view of the documents.

3 Sentence Charts and Extraction of Classification Features

From a practical point of view, it might be reasonable to assume that all information needed to ascertain relevance of a sentence with respect to a query can be reduced to the wording of the sentence (and possibly its context). Given a suitable training set, word n-gram models have been shown to provide reasonable performance for various NLP tasks as text categorization (Schapire and Singer, 2000; Joachims, 1998), calltype classification (Haffner et al., 2003), named entity detection (Levit et al., 2004), question answering (Hakkani-Tür and Tur, 2007) and other areas of speech and language processing. However, since training resources for distillation are indeed very limited, we expect deeper annotations and higher-order dependencies to add valuable (meaning-related) information while abstracting from a particular lexical realization.

Our latest system integrates the following annotation layers:

- **lexical information**: words that comprise sentences (including punctuation, where available)

- **part-of-speech tags and syntactic parses**: we use the Charniak parser (Charniak, 2000) to obtain them
- arguments and targets of **predicate-argument structures** from PropBank (Kingsbury et al., 2002): the Assert parser (Pradhan et al., 2004) is used. PropBank annotations are traditionally syntax-based and focus on tagging of verbs and their arguments and modifiers. For instance, in the sentence *"Yesterday, we saw a fox"* verb *"saw"* takes subject argument (arg0) *"we"*, direct object (arg1) *"a fox"* and is provided with temporal modifier (argm-tmp) *"yesterday"*. Note that not all sentences will have predicate-argument structures found in them. For instance, the verb "be" does not contribute to PropBank annotations. In general, even if for some reason one or several annotation layers cannot be produced for a sentence (for instance, due to deficiencies of the employed parsers), our classification algorithm will remain robust, employing only the features extracted from the present annotation layers.
- **IE elements**: we use the NYU Jet tool (Grishman et al., 2005) to produce ACE annotations as specified in (LDC, 2005). These include entities, relation and event arguments, but also values and timex2 mentions. ACE categorizes IE elements by type (e.g. entity type org stands for organizations) and subtype (e.g. per-family for personal family relations between two entities); it extracts their arguments (e.g. events of subtype conflict-attack can have arguments attacker and target). Furthermore, each annotated element can have several *mentions*, in which case one speaks of pronominal (e.g. *he* for *George W. Bush*) and nominal (e.g. *the president* for *George W. Bush*) coreference.

Next, we explain how the above information types are integrated into one coherent structure that we call a **chart**. While lexical, syntactic, and semantic information has been combined for various language understand tasks, as in the MIT TINA system (Seneff, 1992) or BBN question answering system (Weischedel et al., 2004), charts provide a compact and efficient finite state machine (FSM) representation capturing a wide variety of annotation layers along with query specific slots. A chart can be understood as an amalgam of elements from diverse annotation layers represented as a directed acyclic graph. A chart of a sentence with N words consists of N+1 states that are connected by arcs labeled with **chart entries**. Chart entries can be of several types that reflect the participating annotation layers. Words constitute the simplest, lexical type of chart entry: a trivial chart is just a linear graph whose arcs are words (in their natural order). For instance, in the sentence *"John gave Mary his car."* the word *"John"* gives rise to a lexical chart entry *John*. This chart entry is represented in the sentence chart by an arc that is labeled accordingly and connects state 0 with state 1. To integrate additional annotation elements in a chart, one determines their word spans and connects states corresponding to the first and last words with new arcs that represent the added elements and signify their types. Thus, in the example above, the arc representing syntactic chart entry vp connects state 1 with the state 5. The FSM representation of a chart for this sentence that involves all annotation

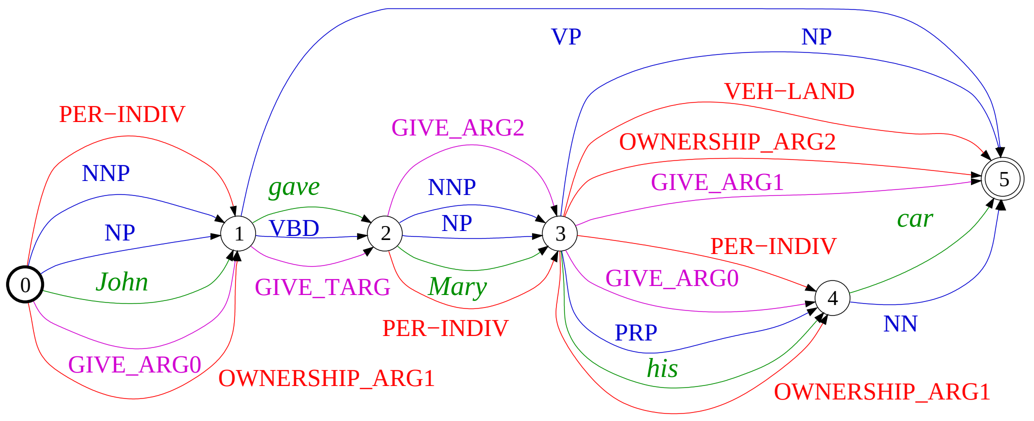

Fig. 2. Chart for the sentence *"John gave Mary his car"*; it contains 5 words (*John, gave, Mary, his* and *car*), 9 syntactic nodes (3x NP, 2x NNP, VBD, PRP, VP and NN), 4 mentions of 3 entities (3x PER-INDIV and VEH-LAND), 3 arguments of an OWNERSHIP relation (2x OWNERSHIP_ARG1 and OWNERSHIP_ARG2) and, finally, target and 3 arguments of verb *give* (GIVE_TARG, 2x GIVE_ARG0, GIVE_ARG1 and GIVE_ARG2).

layers is depicted[4] in Figure 2. For instance, the word phrase *"his car"* gives rise to a syntactic node NP, a mention of an ACE entity VEH-LAND (land vehicle), a second argument of ACE OWNERSHIP relation and, finally, argument ARG1 of a PropBank predicate *gave*.

Charts are similar in spirit to Syntactic and Semantic Graphs (SSGs) in (Hakkani-Tür et al., 2005), but incorporate more information and, because of the integrated coreference (that we obtain from IE annotations as specified by LDC (2005)), allow for more power in modeling sentence meaning.

3.1 Instances of Query Slots as Chart Entries

As illustrated in Eq. 3, query slots represent an important source of classification features. For instance, for the template *"Describe the prosecution of* [PERSON] *for* [CRIME]*."*, the best we can do without paying attention to the information in the slots (i.e. who is the person and what is the crime) is to train a text classifier that selects sentences about prosecution of someone for something. Even if we assume that upstream document retrieval lets through only relevant documents, we are still likely to encounter documents in which various people are being prosecuted for various crimes and, as a result, many false positives should be expected from such a system.

One intuitively straightforward strategy for dealing with query slots is to integrate them in the sentence charts as a separate chart entry type. Each time the textual representation of the i^{th} slot is instantiated in the sentence, a new entry of type $Slot_i$ is introduced in the chart of this sentence. The instantiation process must be flexible

[4] The final period of a sentence is ignored here for the sake of readability.

enough to account for the phenomena like coreference, synonyms and others.

We use this strategy only for query slots that are of types that can be associated with ACE entity types (LDC, 2005),[5] and the instantiation is performed only in sentence intervals that contain entities of such types. For instance, in the case of prosecution of [*Saddam Hussein*], only sentence intervals where ACE annotations indicate the presence of a (non-pronominal) mention of an entity of ACE type PER (person) will be considered for direct instantiation. In addition to the span of the mention itself, we consider spans of its *mention head* (LDC, 2005) and possibly of the syntactic nodes that cover it. We decided to ignore pronominal mentions because coreference information about them tends to be unreliable.

Some entity names may have synonyms or allow for a number of legitimate or tolerable variations. For instance *"New York State"* can be referred to as simply *"New York"*, or as *"Empire State"*. A typo in a newspaper can make it spell *"New York State"* and an ASR misrecognition can turn it into *"New York Slate"*. On the other hand, world knowledge and, in particular, geopolitical relations allow for application-specific synonymy. For instance, if the query is about New York State and an event happened in Albany, we should be able to instantiate the first in the second. To account for variations like this, our instantiation process uses several mechanisms to produce possible alternatives for each name as it is formulated in a query slot. These mechanisms include

- Levenshtein edit distance (to account for typos)
- *Metaphone* algorithm (Philips, 1990) (to model ASR errors)
- gazetteers and other dictionaries (see the Albany-example above)
- *modifier* words: for each of the most common ACE named entities (LDC, 2005), we defined a list of words and expressions that, when used together with the names of this type, are not likely to change their meanings. For instance, one such modifier for names of type GPE (geo-political entity) is *"city of"*. As a result, we consider names like *city of San Francisco* and *San-Francisco* synonyms. There are about 160 such modifier words in our system.
- WORDNET (for deriving base forms and retrieving synonyms)

We recently extended this list by machine learning algorithms as described in (Gillick et al., 2008).

Figure 3 illustrates instantiation of slot *"U.S. President George W. Bush"*. First we recognize *"George W. Bush"* as a *mention head* of an ACE entity PER that covers the slot. Then, in the list of name synonyms we find *"George Bush"* as an alternative for original slot formulation (treating *"W."* as a modifier for personal names), which allows first instantiation. Finally, through coreference information from sentence parses returned by JET (Grishman et al., 2005), we find two other instances (*"Bush"* and

[5] For other slot types such as EVENT and CRIME, alternative mechanisms must be employed (see Section 3.3).

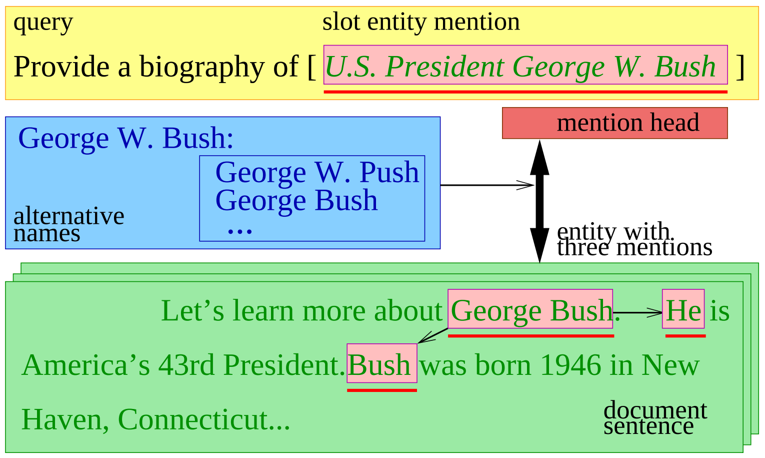
query
slot entity mention
Provide a biography of [U.S. President George W. Bush]
mention head
George W. Bush:
George W. Push
George Bush
...
alternative names
entity with three mentions
Let's learn more about George Bush. He is America's 43rd President. Bush was born 1946 in New Haven, Connecticut...
document sentence

Fig. 3. Instantiation of slot PERSON in a sentence; here, the following steps must be taken in order to find all three instances: a) find head of the slot entity mention, b) use lists of alternative names, c) resolve coreference for sentence entities.

"he") of the slot in the document sentences.

In summary, slot instantiation can be expressed via textual instantiations of one word sequence in another. This kind of matching can be either restricted to be exact or (in a more robust version) understood as a weighted bag-of-words matching. Let $S = s_1 s_2 \dots s_J$ be the word sequence (on the slot side) to be instantiated in the word sequence T (on the candidate sentence side). Then, instantiation score for S in T lies between 0.0 and 1.0 and is computed according to

$$\sigma(S) = \frac{\sum_j I(s_j, T) w(s_j)}{\sum_j w(s_j)}, \tag{4}$$

with the indicator function $I(s_j, T) = 1.0$ if s_j occurs in T and 0.0 otherwise, and weights $w(s_j)$ that reflect relative importance of individual words s_j in S, so that instantiating *"President Chirac"* in *"Chirac"* will produce a higher instantiation score than instantiating it in *"President"*. We postulate these weights to inversely reflect words' prior probabilities (estimated from some large corpus, for instance, WSJ). Alternatively, one can compute them as *Inverse Document Frequencies* of the words. Next, we explain how classification features can be extracted from charts.

3.2 Classification Features from Charts

Once all annotation elements have been introduced into charts, from an operational point of view they become indistinguishable. This means that we can work with combinations of chart entries even if these entries originate from different annotation layers. In particular, we suggest looking at arc labels in the charts and extracting n-grams and *inclusions* thereof as described below.

3.2.1 n-gram Features

Heterogeneous n-grams can be extracted from a chart as intervals of valid paths through its graph representation. For instance, from the chart in Figure 2, the following bigrams and trigrams can be extracted (we use "$\uplus$" as a connector in n-gram representations):

> PER-INDIV $\uplus$ GIVE_TARG $\uplus$ *Mary*
> *gave* $\uplus$ PER-INDIV $\uplus$ NP
> *John* $\uplus$ VP
> PER-INDIV $\uplus$ VP
> VBD $\uplus$ GIVE_ARG2 $\uplus$ GIVE_ARG1
> . . .

The advantage of mixed n-grams (and mixed inclusion from the next section) lies in offering several levels of specificity for each part of a document sentence. It is up to a particular application to decide what words and word sequences it can consider at higher abstraction levels (e.g. syntactic categories). For instance, if the question is about *John*'s actions, then the third feature in the list above seems most relevant. If we are asked about what was given to *Mary*, then the first feature will probably suit our needs better. Finally, if the focus is on the process of giving, then the last feature in the list should be favored with respect to the others. For an example in the context of GALE, consider the query about arrests of members of an organization, and the following highly relevant 3-grams: "ARREST_ARG0 $\uplus$ *arrested* $\uplus$ PER-INDIV" and "ARREST_TARG $\uplus$ SLOT $\uplus$ MEMBERSHIP_ARG1".

The number of n-grams present in a chart like the one in Figure 2 grows exponentially with n. It is also polynomial in the number of annotation layers and linear in the sentence size. This magnitude of potential features can be paired with a task-dependent feature selection process, in order to keep only those features that strike the right balance between specificity and generality, while ignoring everything else. The work of selecting the most effective features from the large pool of available ones could be accomplished with traditional feature selection mechanisms such as information gain (Quinlan, 1986) or it could be performed by humans who would use their language and domain competence to explicitly distinguish those n-grams deemed most relevant for the task. Due to computational constraints, we only consider uni-, bi- and trigrams in our experiments.

3.2.2 Inclusion Features

While n-grams contain information about sequences of chart entries, the inclusion features focus on parallelism. From the chart in Figure 2, for example, we can see that the word *John* coincides with a syntactic node NP and with an ACE entity of type PER-INDIV (person, individual), or that the word *"car"* constitutes a part of the second argument of the OWNERSHIP relation and ARG1 of the “giving” predicate. In other words, inclusion features focus on pairs of chart entries that contain (“include”) each other in terms of word spans. Currently, our system makes use only of pairs where the shorter entry covers at least 30% of the longer one. In the chart in Figure 2, some of the inclusion features are (“$\ni$” stands for “includes”)

OWNERSHIP_ARG1 $\ni$ PER-INDIV
S $\ni$ VP
VEH-LAND $\ni$ *his*

The significance of the inclusion features becomes more apparent if we modify our example to read *“My neighbor John, someone told me, gave Mary his car.”* and ask about John’s actions. Due to a lacking adjacency between the informative chart entries describing John and his actions, the meaning of the sentence will not be captured by *n*-grams (and generally by gappy *n*-grams neither). However, the inclusion features will, for instance, provide us with the relevant information about the slot (*John*) being a part of the subject argument in the “giving” predicate.

3.2.3 Real Valued Extensions

While it would suffice to keep *n*-gram and inclusion features binary (thus, only indicating their presence in the charts), we discovered that upgrading these features to real-valued domains is beneficial for the classification task. First, we assign scores to all chart entries determined by (a) relative salience of the words they consist of and (b) uncertainty of their instantiation (see, for instance, the next section discussing instances of query slots). Then, values of *n*-gram features would be computed as the minimum of scores of all participating chart entries. In addition, for inclusion features, coverage fraction acts as an extra multiplicative factor in the formula.

3.3 Accounting for non-ACE Slots

In Section 3.1 we explained how query slot types that have analogies among ACE entity types (LDC, 2005) can be integrated in sentence charts, and through these charts participate in the feature generation process. In the framework of the GALE project (BAE, 2006) which is the focus of this research, slot types PERSON, ORGANIZATION, COUNTRY, and LOCATION satisfy this condition. However, this is not the case for other slot types such as TOPIC, EVENT, and CRIME which are typically expressed with greater lexical variability (e.g., EVENT=*“Looting of Iraqi museums after U.S. invasion”*). It is difficult or even impossible to instantiate such slots at any particular location because their lexical form may be very different from the slot formulation and they can be spread over an entire sentence or even several sentences of a document. For instance, for event formulation *“attacks in Indonesia”*, the two sentences *“There was terrible loss of lives in Bali. Two hundred people died in a bomb blast.”* are perfectly on-topic even though they do not contain a single content word from the original event formulation. Instead, we can try to estimate *topicality* of a sentence with respect to a topic. In (Levit et al., 2007b) we introduced an approach that estimates topicality based on lexical and structural similarities of proposition trees extracted for sentence and topic description. In the present approach we opt for a lightweight version of topicality that, while lacking the extent of the structural knowledge of (Levit et al., 2007b), takes advantage of IE and other annotations.

Much like we create charts for document sentences, we can also create them for wordings of query slots. Then, by comparing slot and sentence charts in terms of the chart entries they contain, we can generate a number of topicality features. Each of these features reflects the average extent to which chart entries of a specified category that are present in the slot can also be instantiated in the sentence. The following topicality categories are supported:

(1) words
(2) ACE entities of types PER, ORG, and GPE&LOC (with and without instantiations across different entity types)
(3) ACE relations (with and without cross-type instantiations)
(4) ACE events (with and without cross-type instantiations)
(5) targets of PropBank predicates

As an example, think of the topic formulated as *"Friendship of John and Peter"* and the sentence that we used in Figure 2 and assume that *"John"* and *"Peter"* were both correctly labeled as entities of type/subtype PER-INDIV in the slot parse. Then, PER-INDIV entity *"John"* from the slot has a perfect instance in the sentence (its score in terms of an instantiation mechanism like the one from Section 3.1 is 1.0), while PER-INDIV entity *"Peter"* has no instances (score 0.0). Then, the value of the topicality feature TOPICALITY-ENTITY-PER for this slot in this sentence is set to be $(1.0 + 0.0)/2 = 0.5$. When instantiating ACE relations from slots, we aggregate instantiation scores computed separately for both relation arguments, while also penalizing mismatches in relation types and subtypes. A similar procedure is used for ACE events.

While deciding on the topicality of a sentence, it is also important to take into account sentences in its vicinity. Therefore, classification features representing each sentence should include not only its chart-based features and its topicality features, but also topicality features of the sentences in its direct neighborhood (context).

3.4 Feature Extraction Revisited

A diagram in Figure 4 summarizes all types of classification features that our system extracts and uses for classification. The diagram shows how classification features are computed for each sentence. First, a chart is created for the sentence. Then the chart

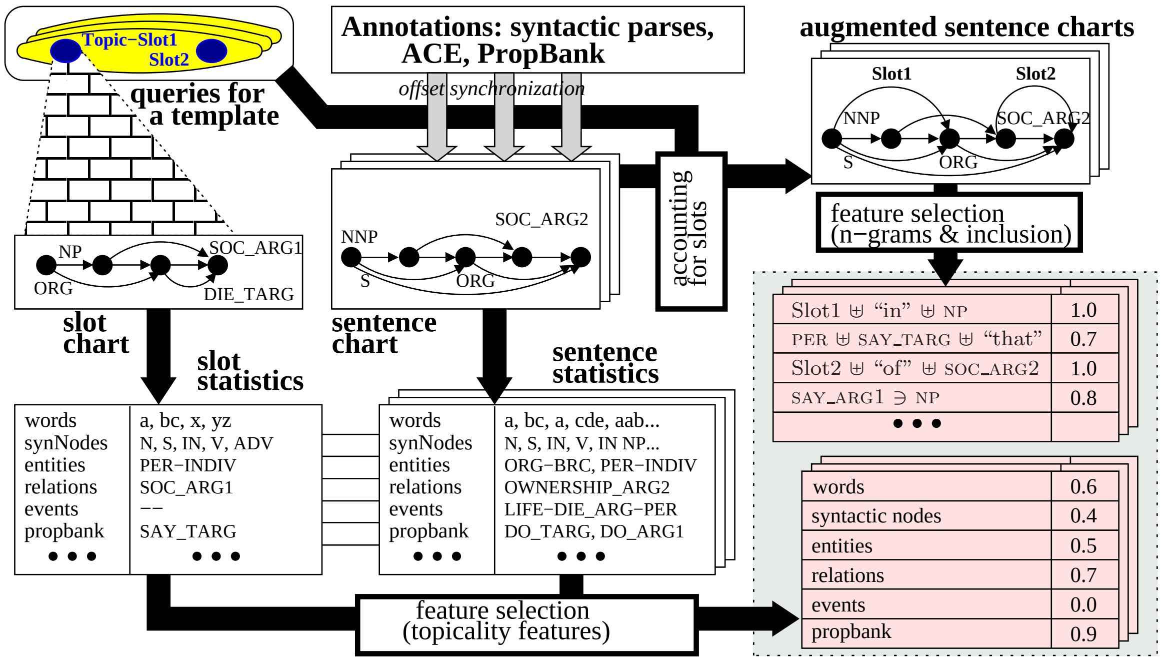

Fig. 4. Extraction of classification features from sentences

is augmented with the slot information from the query. From this augmented chart, a vector of n-gram and inclusion features and their values is extracted. Charts are created for the query slots as well. By comparison of these charts with the original sentence chart, a vector of topicality features is built and concatenated with the vector above to form the final feature vector used for training and classification. While the number of topicality features is known in advance, the pool of n-gram and inclusion features that can be used for classification, must be created based on the training sentences; features extracted from each test sentence are then pruned accordingly.

4 Experiments and Results

4.1 Data Sets and Experimental Setup

All the experiments in this paper are conducted on GALE data. We considered five query templates that are shown in Table 1. For each of the templates in this table, several training and test queries have been provided along with *answer keys*: relevant sentences (with document IDs) selected by humans. These are then converted to labels "relevant" and "irrelevant" for each sentence in the affected documents. For our experiments we used documents that (according to the labelers) contain at least one relevant sentence. While not suitable for an unbiased evaluation of the end-to-end distillation system, this restriction represents the best way to perform a standalone evaluation of the sentence extraction algorithm that is the main focus of this paper.

templ.	formulation
T1	List facts about event: [EVENT]
T8	Describe the prosecution of [PERSON] for [CRIME]
T12	Provide a biography of [PERSON]
T15	Identify persons arrested from [ORGANIZATION] and give their names and roles in the organization and time and location of arrest
T16	Describe attacks in [LOCATION] giving location, date and number of dead and injured

Table 1
Query templates considered for system evaluation.

	corpus	queries per template	docs	sents	avg.# rel. sents. per query	avg.# irrel. sents. per query
training	GALE-Y1	24	2719	38003	72	264
test	GALE-Y2	9	507	15529	39	299

Table 2
Selected statistics for training and test corpora.

Documents that supplied answer keys for training and test queries came from different corpora. To train our classifiers, we used the English text part of the GALE Y1 training data, and the models were then tested on the English text part of the GALE Y2 corpus. Some of the corpus statistics, including average number of queries per template, number of documents and sentences that provide support for these queries, as well as the average number of relevant and irrelevant sentences per query, are shown in Table 2. Note that the test corpus has a significantly smaller proportion of relevant snippets (11% as opposed to 21% in the training corpus) and, moreover, its queries were judged as subjectively more challenging. With some 3K sentences per template and 15K in total in the test set, the F-measure improvements will be statistically significant at $\alpha = 0.05$, if they are no less than 0.03 and 0.01 respectively.

All our experiments are conducted using the Support Vector Machine classifier SVM^{light} (Joachims, 1999) and organized as follows: after extracting classification features for each template, we first use training queries to perform leave-one-out optimization of the rejection threshold in the dichotomy "relevant" vs. "irrelevant". Then, a new model is trained using all training queries of the template. This model, along with the optimized rejection threshold is then applied to the test queries, using F-measure (uniformly weighted harmonic mean of precision and recall) with respect to the class of the relevant sentences as a figure of merit for evaluation.

exp#	template	T1	T8	T12	T15	T16	overall
0:	accept-all	0.40	0.32	0.17	0.24	0.33	0.29
1:	words	0.42	0.41	0.24	0.45	0.46	0.39

Table 3
Baseline classification results (F-measure) for a) accept-all trivial classifier, b) word bigrams.

4.2 Annotation Layers, Feature Types and Classification Results

We now present experiments designed to illustrate how information distillation can benefit from additional annotation layers (through corresponding chart entry types) and how different types of features influence classification results. Tables 3 through 6 show how augmenting sentence charts with new layers in addition to lexical entries improves classification performance. Also, the influence of different feature types is elucidated. In Table 3, performance of two baselines is presented: the first one ("accept-all", exp#0) shows how a trivial classifier that marks all sentences as "relevant" (perfect recall / low precision classifier) performs on the test data. The high F-measure of this trivial classifier for template T1 is due to the "oracle" IR that we employed in our experiments (see Section 4.1). Unlike other templates, with T1, training and test documents are often dedicated entirely to the event in question, so that most of the sentences in them are on-topic (e.g., for event *"Civil unrest in France"* this proportion is about 65%). The second baseline, exp#1, represents a slot-ignorant classifier that considers only word bigrams to make predictions. One can consider this baseline as a trivial case of extracting bigrams from charts that lack any information other than lexical. As expected, this classifier achieves the best improvement for those templates (arrests, attacks and prosecution) that can be associated with specific words.

Table 4 follows improvement of classification results as we augment charts with entries from other annotation layers and only then extract bigram classification features. For instance, as we add IE-based chart entries, and thus allow for "mixed" bigram features such as "PER-INDIV $\uplus$ *gave*", the overall F-measure increases from 0.39 to 0.41 (exp#2). We see that different features turn out to be sensible for different templates. While attempts can be made to explain some of these observations (for instance, the biography template is expected to be more responsive to query slots instantiated in the sentences than the arrest-template, as there are many topics considered relevant for people's biographies while arrests do imply a very specific subset thereof), in general, we believe that it is safer to accept the fact that there will be template-specific variation in the optimal feature sets. We also observe, however, that in our experiments the best results have been achieved either by taking advantage of all available annotations or by using words and slot chart entries (whose instantiation, as described in Section 3.1, already involves other annotation layers). One exception are PropBank features that do not seem to help classification for any of the templates. In part, this might be caused by the explicit focus on verbs that we observe for

exp#	template	T1	T8	T12	T15	T16	overall
1:	words	0.42	0.41	0.24	0.45	0.46	0.39
2:	words+IE	0.43	0.40	0.28	0.48	0.46	0.41
3:	words+syntax	0.44	0.41	0.27	0.50	0.48	0.42
4:	words+propbank	0.43	0.40	0.23	0.37	0.47	0.38
5:	words+slots	0.42	**0.51**	**0.45**	0.44	0.48	0.46
6:	words+all	**0.45**	0.46	0.42	**0.51**	**0.52**	**0.47**

Table 4
Classification with bigram features from augmented charts.

exp#	template	T1	T8	T12	T15	T16	overall
7:	words+IE	0.41	0.41	0.26	0.34	0.44	0.37
8:	words+syntax	0.41	0.33	0.23	0.43	0.46	0.37
9:	words+propbank	0.41	0.38	0.22	0.36	0.45	0.36
10:	words+slots	0.40	**0.51**	0.40	0.40	0.47	0.43
11:	words+all	**0.43**	0.48	**0.42**	**0.45**	**0.48**	**0.45**

Table 5
Classification with inclusion features from augmented charts.

this feature family, while many of the queries are formulated using nominalizations; compare, for instance: *"attacks in Iraq"* (query) vs. *"in Iraq insurgents attacked a military base"* (candidate sentence).

Table 5 is concerned with using only inclusion features extracted from charts, instead of the bigram features. First, we see again that the best classification results are achieved with all annotations or only with words and slots (for biography queries). In direct comparison, inclusion features provide less discriminative information to the classifier; on the other hand, their number amounts to only 40% of the number of the bigram features, which speeds up classification.

Table 6 brings inclusion and n-gram features together[6] in exp#12, but also explores the contributions of topicality features from Section 3.3. It appears that inclusion features, while helping certain templates, do not change the overall picture, and the results of the combination (exp#12) are similar to what bigrams deliver alone (exp#6). As far as topicality features are concerned, their impact is very significant (exp#13) and, for all templates except T8, is the most powerful when topicality features are considered with other features (exp#14). The last row in the table (#exp15) shows

[6] To render presentation more illustrable, we use notation "X+Y" to signify that classification features from experiments X and Y are pulled together.

exp#	template	T1	T8	T12	T15	T16	overall
12:	6+11	0.43	0.48	0.44	0.46	0.54	0.47
13:	words+topic	0.45	0.57	0.43	0.40	0.49	0.46
14:	12+topic	**0.47**	0.52	**0.55**	**0.48**	0.53	**0.51**
15:	14, no context	0.47	0.51	0.52	0.46	0.50	0.49

Table 6
Classification with inclusion features from augmented charts.

the importance of considering sentence context (three sentences before and one sentence after the candidate sentence) while producing topicality features as suggested in Section 3.3. Most of the templates agree that taking context into account for topicality features helps classification (ironically, for T1, where the impact of context was expected to be the highest, it turned out to be the smallest).

Figure 5 summarizes the results above, showing individual performance of each query with the trivial "accept-all" baseline, the baseline of word bigrams, and the final system that uses all available features (also bigrams). In addition, this plot illustrates how many sentences there were to classify for each query. Notice that the queries where the trivial baseline is better are usually fairly small in terms of the test set, and comprise documents that are centered around the event/topic/subject in question. Nonetheless, we decided to compute a template-wise F-measure by uniformly averaging F-measures achieved on all queries of a template instead of deriving it from pooled sentence classification statistics. In this way, we can predict the algorithm's performance on an unseen query instead of an unseen sentence, and do not need to worry about the highly skewed distribution of query "sizes". The plot shows clearly that biographical queries (template 12) tend to profit the most from the non-lexical features in the system, while a significant improvement for topic-based queries (template 1) requires more effort.

4.3 Effect of Name Alternatives

In Section 3.1 we explained that it is important to be flexible while instantiating names from query slots in sentences and suggested several ways of how these names can be augmented with their equivalents. Now we justify this decision experimentally. The easiest and most meaningful way of doing so is to see what happens to exp#5 (where classification features are bigrams of words and slots) when we exclude name alternatives from the algorithm. Doing just that, we observed the average F-measure dropping from 0.46 to 0.43. We can observe a similar effect with the final system (exp#14) where the overall F-measure decreases from 0.51 to 0.50 (significant at level $\alpha = 0.05$) when alternative names are dropped. For more information on our current algorithms for discovering name alternatives see (Gillick et al., 2008).

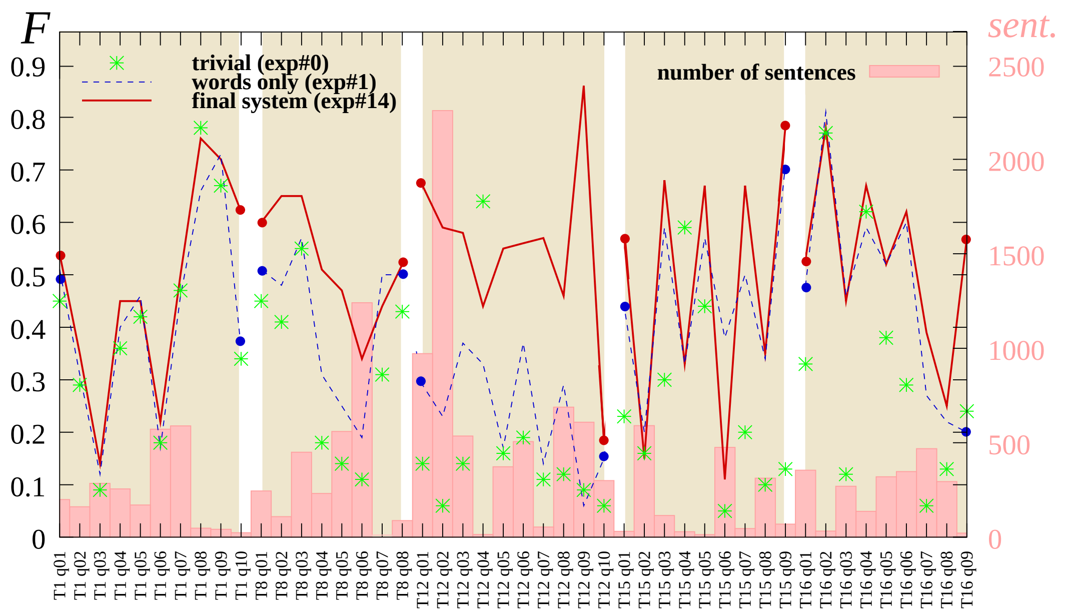

Fig. 5. Comparison of F-measure for baselines exp#0, exp#1 and final system exp#14 for each test query of all five templates. In query notation Txqy, x stands for the template number and y for the query number within this template. The number of test sentences for each query is provided using the scale on the right.

	n=1		n=2		n=3	
exp#	#features	F	#features	F	#features	F
1	$9.7 * 10^3$	0.37	$7.8 * 10^4$	0.39	$2 * 10^5$	0.41
14	$1.3 * 10^5$	0.49	$4.3 * 10^5$	0.51	$2.6 * 10^6$	0.49

Table 7
Effect of maximum n-gram size on classification for a) word-only system, b) final system.

4.4 Effect of n-gram Size

A natural question to investigate for our algorithm is the maximum length of n-grams as classification features. What happens, for instance, if we consider trigrams of words as opposed to mixed bigrams? Table 7 answers this question. It shows average F-measures achieved by unigrams, bigrams, and trigrams in a words-only setting as well as in the final constellation as in exp#14.

The first row of the table shows that when restricted to words-only features, trigrams tend to outperform bigrams. However, in the final setting (second row), bigrams deliver a significantly better performance than not only unigrams (which naturally possess smaller discriminative power), but also trigrams. The problem with trigrams here seems to lie in the sheer number of the generated sparse features. Given this outcome we plan to direct our future investigations toward intelligent feature selection. Most important yet, our final system, even when based solely on unigrams, is clearly better than word trigrams (F=0.49 vs. F=0.41).

4.5 Filtering Sentences with Handcrafted Rules

In (Levit et al., 2007a) we indicated that combining a statistical approach with rule-based filtering can lead to further classification improvements. In particular, for each query template we selected a number of classification features that had to be present among the features generated for a candidate sentence, or the sentence will be assumed irrelevant by default. As we applied the same technique to exp#14 in the present paper, the rule-based filtering led to an additional three percent absolute increase of the overall F-measure (0.54).[7]

5 Related Work

Question answering (QA) has a rich history of previous research. At the latest when TREC-8 introduced QA-track (Voorhees, 1999) to extend and refine existing document retrieval (DR) functionality, sentence-level question answering stepped into the spotlight. Since then, several commercial and government-sponsored programs appeared that are primarily concerned with finding pinpoint answers to user queries from large collections of documents (e.g. Voorhees, 2003; Magnini et al., 2003; BAE, 2006; Warthen, 2000; Dang, 2005).

While some authors reported on successful QA systems that use only lexical pattern matching (Soubbotin and Soubbotin, 2001; Ravichandran and Hovy, 2002), the general consensus in the field is that much more can be achieved with deeper analysis of documents and queries. Blair-Goldensohn et al. (2004), for instance, manually designed a number of lexico-syntactic patterns to identify *definitional predicates* that are later combined using summarization techniques to form answers to definitional questions. Syntactic analysis of queries has been a part of many QA systems as well (see, for instance, Katz and Lin, 2003).

Being more robust and meaning oriented than syntactic parses, semantic labels have also been a subject of QA-related investigations. In (Levit et al., 2007b) it was shown how semantic relations inferred from syntactic trees and expressed as predicate-argument structures (*propositions*) can be employed to deal with open-domain templates such as *"Describe the facts about [*EVENT*]"*. In particular, because of their nearly absolute lexical coverage, propositions extracted from slot formulations can be organized into *proposition trees* and instantiated in proposition trees extracted from sentences. One of the algorithms described in (Kaisser and Webber, 2007) parses questions

[7] Differences between these and previously reported F-measures are due to cleaned data, different parameter settings, and improvements to the actual algorithm.

with semantic roles as defined in PropBank (Kingsbury et al., 2002), FrameNet (Baker et al., 1998) or VerbNet (Schuler, 2005) and, based on these parses (and in the case of FrameNet also on inter-frame relations for the head verb), suggests several alternative queries to a web search engine whose responses are then parsed with the same resources and the results scored against the structure of the expected answers. A similar approach that uses web search engines and relies on matched semantic role labels to find answers to factoid queries is described in (Stenchikova et al., 2006). In the related *Textual Entailment* task that, given two text fragments, requires to recognize whether the meaning of one text is entailed (can be inferred) from the other text (Dagan et al., 2006), deep semantics are extensively utilized as well (e.g. Tatu and Moldovan, 2005; de Marneffe et al., 2006a). Tatu and Moldovan (2005) report on a system that parses sentences extracting 18 types of *semantic relations* from them and then uses *semantic axioms* to generate secondary relations. After that, the system looks for a contradiction between examined text and a negated hypothesis using COGEX logic prover (Moldovan et al., 2003) and assumes entailment if none is found.

Information extraction elements have traditionally been a reliable partner for QA systems. In (Srihari and Li, 1999), named entities and other IE elements (reminiscent of the later-introduced ACE annotations) were used to constrain the scope of candidate sentences to those containing the IE elements associated with a query. The associations were established through a number of handwritten rules that operated on structural and lexical representations of the queries. More recent publications prefer IE elements from ACE annotation guidelines that are defined by the annual NIST ACE evaluations (LDC, 2005) and include, among other tasks, mention coreference resolution (e.g., pronominal coreference), entity extraction (for people, organizations, locations, and so on), and entity relation and event extraction (e.g., ownership relation and attack event). For instance, in (Schiffman et al., 2007), the presence of relevant ACE events and ACE entities in the vicinity of a sentence is taken as an indicator of the sentence's relevance. In our own previous work (Hakkani-Tür et al., 2007) we recognized the advantage of restrictive rules involving ACE elements as well.

Numerous successful attempts to combine the annotations above have been made. The system described in (Weischedel et al., 2004) answers biographical questions by combining predicate-argument structures and IE elements to generate *features* that are used to predict relevance of a sentence via the Bayes rule with conditional probabilities estimated from known biography sources. In (de Salvo Braz et al., 2005) questions and sentences are converted into hierarchies of lexical representations, syntactic parses, semantic roles and named entities (called *concept graphs*), and compared via subsumption, while constantly re-representing underlying texts using rewrite rules (such as *"Z's W⟶W of Z"*). While the idea of compiling different annotation layers into one graph representation has already been suggested for annotation purposes (Bird and Liberman, 1999), applied to a classification task as performed by our system, it represents a novel approach as the "tight" integration happens already at the stage of feature generation.

Several of the systems described above and many others use inference rules to induce new facts/relations from the available ones. The COGEX logic prover from (Moldovan et al., 2003) builds upon logical representations into which questions and potential answer sentences are transformed, and uses first-order logic to produce an inference between them. World knowledge and natural language processing axioms are employed to facilitate derivation. An application of probabilistic MAP (maximum a posteriori) inferences to question answering is detailed in (Narayanan and Harabagiu, 2004). The method exploits the fact of presence of some semantic relations (such as semantic roles in PropBank) in a sentence to conclude on applicability of a parametrized event representation. We look forward to introducing reasoning in IXIR as well and plan on achieving it through integrating FrameNet annotations.

Finally, regarding the system described in this paper, we had shown in (Levit et al., 2007a) that it can be applied not only to written English texts, but to ASR output as well.

6 Conclusion and Future Work

We presented a system for information distillation that extracts relevant sentences for templated queries with variable slots. The system introduces a novel model for generating classification features. We extract n-grams and *inclusion* features from sentence charts which are directed acyclic graphs of word-aligned elements from various annotation layers (such as lexical representations, syntactic parses, semantic role labels and ACE annotations), and also compute sentence *topicality* features as the relative amount of information from a query slot that can be instantiated in the sentence, whereby topicality for different chart entry types (e.g., words or ACE entity PER) is computed separately. We inspected contributions of individual annotation layers and feature types to overall system performance and determined that bigrams have a better classification potential than inclusion features and that instantiating query slots in the sentences is crucial. Furthermore, we observed that while different templates tend to benefit from different annotation layers, the combination of all features leads to the best overall results, even though the large number of generated features is a bottleneck to be concerned about. In particular, our final system achieved an F-measure of 0.51 (0.54 when augmented by handcrafted rules), which amounts to 31% (38%) improvement compared to F=0.39 of an analogous system that uses only lexical features for classification. The system clearly outperforms the "accept-all" trivial baseline with F=0.29.

In Section 4.4 we observed how a large number of features tends to impede classification in the case of trigrams. Even bigrams seem to struggle with the same problem. Therefore, exploring ways of combining and selecting features became one of the highest-priority issues for our team. We plan to employ several linguistic and information-theoretical criteria to do so. Besides, we are currently looking into using typed dependency parses (de Marneffe et al., 2006b) to augment our feature extraction mechanisms, and will report on the results in future publications. Finally, we are planning to invest more effort into coping with noise due to ASR and machine translation errors.

Acknowledgements

We thank the SRI GALE Distillation team members at Columbia University, University of Massachusetts, NYU, and Fair Isaac. This material is based upon work supported by the Defense Advanced Research Projects Agency (DARPA) under Contract No. HR0011-06-C-0023. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of DARPA.

References

BAE, 2006. Go/No-Go Formal Distillation Evaluation Plan for GALE. Tech. rep. URL `https://www.sainc.com/GALE-BAA/public/`

Baker, C. F., Fillmore, C. J., Lowe, J. B., 1998. The Berkeley FrameNet Project. In: Proceedings of COLING-ACL'98. Montréal, Canada, pp. 86–90.

Bird, S., Liberman, M., 1999. A Formal Framework for Linguistic Annotation. Tech. Rep. MS-CIS-99-01, Philadelphia, PA.

Blair-Goldensohn, S., McKeown, K. R., Schlaikjer, A. H., 2004. Answering Definitional Questions: A Hybrid Approach. In: Maybury, M. (Ed.), New Directions In Question Answering. MIT Press, Ch. 4, pp. 47–57.

Charniak, E., 2000. A Maximum-Entropy-Inspired Parser. In: Proceedings of NAACL-2000.

Dagan, I., Glickman, O., Magnini, B., January 2006. The PASCAL Recognising Textual Entailment Challenge. In: Lecture Notes in Computer Science. Vol. 3944. pp. 177–190.

Dang, H. T., 2005. Overview of DUC-2005. In: Proceedings of the 2005 Document Understanding Conference at NLT/EMNLP.

de Marneffe, M.-C., MacCartney, B., Grenager, T., Cer, D., Rafferty, A., Manning, C. D., 2006a. Learning to Distinguish Valid Textual Entailments. In: Second Pascal RTE Challenge Workshop.

de Marneffe, M.-C., MacCartney, B., Manning, C. D., 2006b. Generating Typed Dependency Parses from Phrase Structure Parses. In: LREC. Genoa, Italy.

de Salvo Braz, R., Girju, R., Punyakanok, V., Roth, D., Sammons, M., 2005. Knowledge Representation for Semantic Entailment and Question Answering. In: Proceedings of IJCAI-05 Workshop on Knowledge and Reasoning for Question Answering.

Gillick, D., Hakkani-Tür, D., Levit, M., September 2008. Unsupervised Learning of Edit Parameters for Matching Name Variants. In: Proceedings of Interspeech'08. Brisbane, Australia.

Grishman, R., Westbrook, D., Meyers, A., 2005. NYU's English ACE 2005 System Description. Tech. Rep. 05-019, NYU.

Haffner, P., Tur, G., Wright, J., April 2003. Optimizing SVMs for Complex Call Classification. In: Proceedings of ICASSP-2003. Hong Kong.

Hakkani-Tür, D., Tur, G., April 2007. Statistical Sentence Extraction for Information Distillation. In: Proceedings of ICASSP-2007. Honolulu, HI.

Hakkani-Tür, D., Tur, G., Chotimongkol, A., June 2005. Using Syntactic and Semantic Graphs for Call Classification. In: Workshop on Feature Engineering for Machine Learning in Natural Language Processing at ACL-05. Ann Arbor, MI.

Hakkani-Tür, D., Tur, G., Levit, M., August 2007. Exploiting Information Extraction Annotations for Document Retrieval in Distillation Tasks. In: Proceedings of Interspeech'07. Antwerp, Belgium, pp. 330–333.

Joachims, T., 1998. Text Categorization with Support Vector Machines: Learning with Many Relevant Features. In: Proceedings of ECML-98. Chemnitz, Germany, pp. 137–142.

Joachims, T., 1999. Making Large-Scale SVM Learning Practical. In: Schölkopf, B., Burges, C., Smola, A. (Eds.), Advances in Kernel Methods - Support Vector Learning. MIT-Press.

Kaisser, M., Webber, B., 2007. Question Answering Based on Semantic Roles. In: Proceedings of the ACL-2007 Deep Linguistic Processing Workshop, ACL-DLP'2007. Prague, Czech Republic.

Katz, B., Lin, J., April 2003. Selectively Using Relations to Improve Precision in Question Answering. In: Proceedings of the EACL-2003 Workshop on Natural Language Processing for Question Answering. Budapest, Hungary.

Kingsbury, P., Palmer, M., Marcus, M., 2002. Adding Semantic Annotation to the Penn TreeBank. In: Proceedings of HLT-2002. San Diego, CA.

LDC, 2005. ACE English Annotation Guidelines for Entities. Tech. rep.
URL `http://projects.ldc.upenn.edu/ace/`

Levit, M., Boschee, E., Freedman, M., August 2007b. Selecting On-Topic Sentences from Natural Language Corpora. In: Proceedings of Interspeech-2007. Antwerp, Belgium, pp. 2793–2796.

Levit, M., Haffner, P., Gorin, A., Alshawi, H., Nöth, E., 2004. Aspects of Named Entity Processing. In: Proceedings of ICSLP-2004. Jeju, South Korea.

Levit, M., Hakkani-Tür, D., Tur, G., December 2007a. Integrating Several Annotation Layers for Statistical Information Distillation. In: ASRU-07. Kyoto, Japan.

Magnini, B., Romagnoli, S., Vallin, A., Herrera, J., Peinado, V., Verdejo, F., De Rijke, M., 2003. Multiple Language Question Answering Track at Third Workshop of the Cross-Language Evaluation Forum. In: Peters, C., Braschler, M., Gonzalo, J., Kluck, M. (Eds.), Cross-Language Information Retrieval and Evaluation. Springer Verlag, Berlin.

Moldovan, D., Clark, C., Harabagiu, S., 2003. COGEX: A Logic Prover for Question Answering. In: Proceedings of HLT-NAACL'03. Edmonton, Canada.

Narayanan, S., Harabagiu, S., 2004. Question Answering Based on Semantic Structures. In: Proceedings of the 20th Int. Conf. on Computational Linguistics (COLING 2004). Morgan Kaufmann, San Francisco, CA.

Philips, L., 1990. Hanging on the Metaphone. Computer Language 7 (12), 38–44.

Pradhan, S., Ward, W., Hacioglu, K., Martin, J., Jurafsky, D., May 2004. Shallow Semantic Parsing using Support Vector Machines. In: Proceedings of HLT/NAACL-2004. Boston, MA.

Quinlan, J. R., 1986. Induction of Decision Trees. Machine Learning 1 (1), 81–106.

Ravichandran, D., Hovy, E., 2002. Learning Surface Text for a Question Answering System. In: Proceedings of the 40th ACL conference. Philadelphia, PA.

Schapire, R. E., Singer, Y., 2000. BoosTexter: a Boosting-based System for Text Categorization. Machine Learning 39 (2-3), 135–168.

Schiffman, B., McKeown, K., Grishman, R., Allan, J., April 2007. Question Answering Using Integrated Information Retrieval and Information Extraction. In: Proceedings of HLT-NAACL'07. Rochester.

Schuler, K. K., 2005. VerbNet: A Broad Coverage, Comprehensive Verb Lexicon. Ph.D. thesis, University of Pennsylvania.

Seneff, S., 1992. TINA: a Natural Language System for Spoken Language Applications. Computational Linguistics 18 (1), 61–86.

Soubbotin, M. M., Soubbotin, S. M., November 2001. Patterns of Potential Answer Expressions as Clues to the Right Answer. In: Proceedings of the TREC-10 Conference. Gaithersburg, MD.

Srihari, R., Li, W., November 1999. Information Extraction Supported Question Answering. In: Proceedings of TREC-8. Gaithersburg, MD.

Stenchikova, S., Hakkani-Tür, D., Tur, G., September 2006. QASR: Question Answering Using Semantic Roles for Speech Interface. In: Proceedings of Interspeech-2006 (ICSLP). Pittsburg, PA, pp. 1185–1188.

Strohman, T., Metzler, D., Turtle, H., Croft, W. B., May 2005. INDRI: A Language-Model Based Search Engine for Complex Queries. In: Proceedings of the International Conference on Intelligent Analysis. McLean, VA.

Tatu, M., Moldovan, D. I., 2005. A Semantic Approach to Recognizing Textual Entailment. In: Proceedings of HLT/EMNLP'05. pp. 371–378.

Voorhees, E. M., 1999. The TREC-8 Question Answering Track Report. In: Proceedings of the Eighth Text REtrieval Conference (TREC-8). pp. 500–246.

Voorhees, E. M., 2003. Overview of the TREC 2003 Question Answering Track. In: TREC. pp. 54–68.

Warthen, D., 2000. Grammar Template Query System. Patent WO 00/57302; World Intellectual Property Organization.

Weischedel, R., Xu, J., Licuanan, A., 2004. A Hybrid Approach to Answering Biographical Questions. In: Maybury, M. (Ed.), New Directions in Question Answering. MIT Press, Ch. 5, pp. 59–69.